

FIGURE 1 (Prior Art)

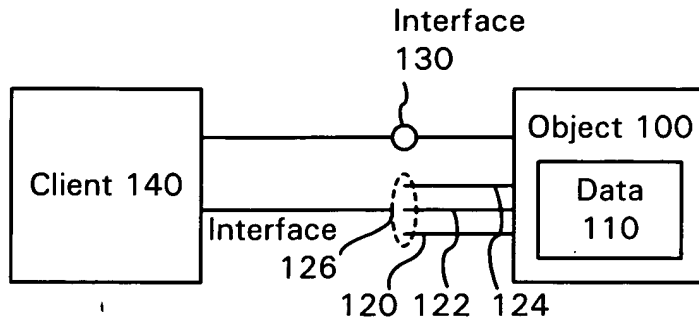


FIGURE 2 (Prior Art)

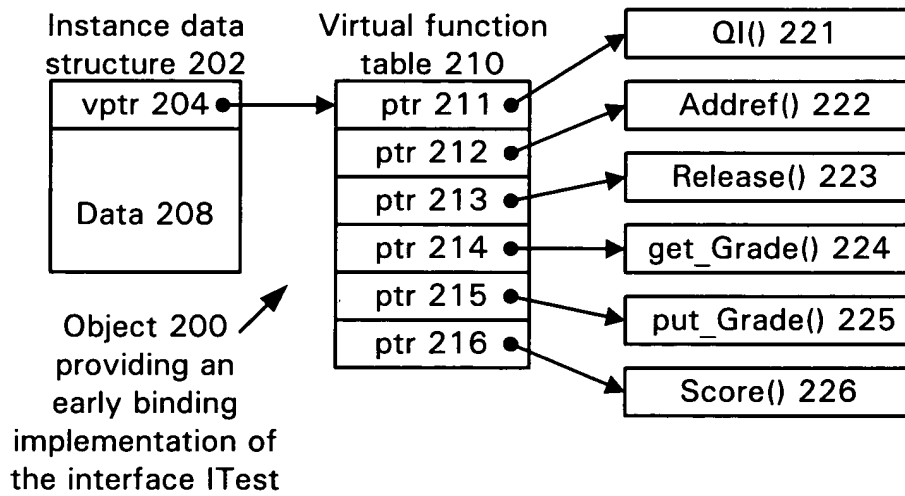


FIGURE 3 (Prior Art)

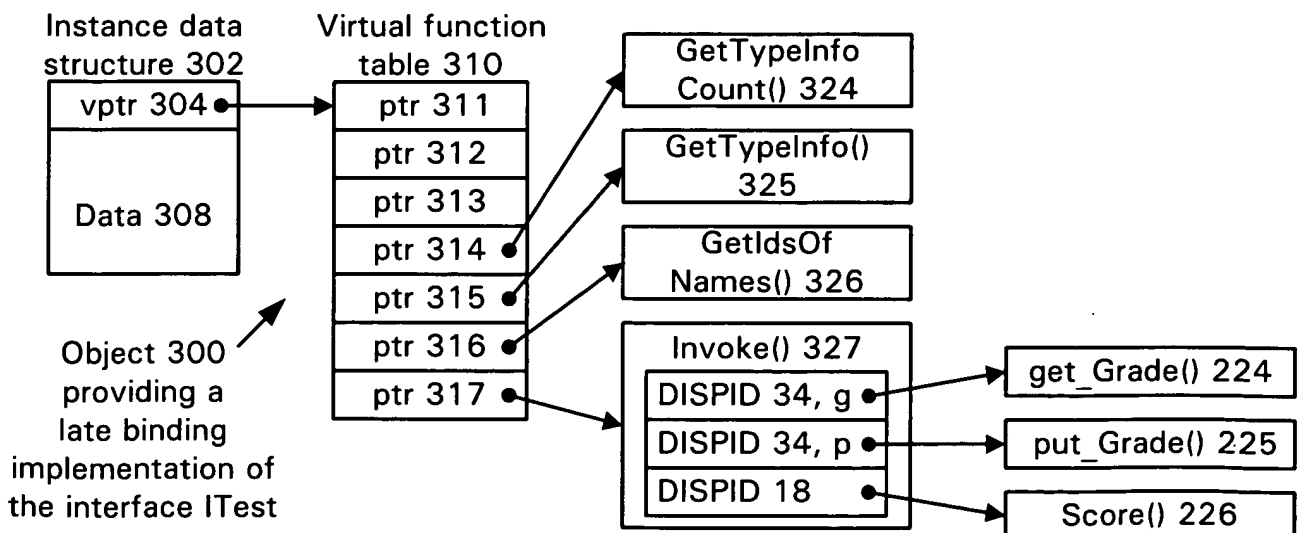
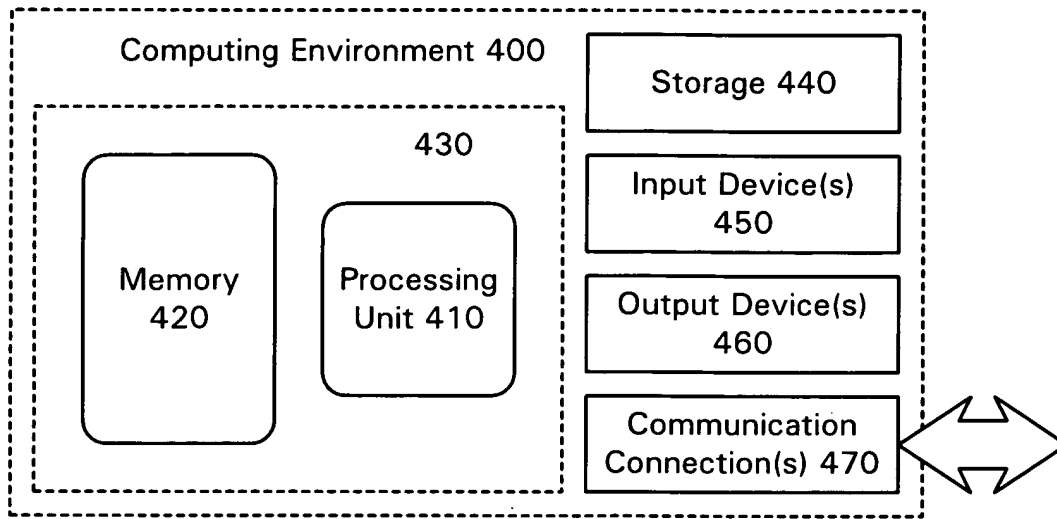


FIGURE 4



009020-2041960

FIGURE 5

```

500 #define _ATL_ATTRIBUTES 1
#include "atlbase.h"
#include "atltcom.h"
extern "C" int printf(const char*, ...);

[project(name = MyLib, helpfile = "test.res", helpcontext = 12)];

[export] enum E {
    e1 = 17
};
[export] struct S {
    int i, j;
};

[dispinterface, helpstring("interface ITest")] __interface ITest : IDispatch {
    [id(34), propput] void Grade([in] E e0, [in] char c);
    [id(34), propget] void Grade([in] E e0, [out, retval] char* pc);
    [id(18)] HRESULT Score([in] S* a, [in] float b, [in] VARIANT c);
};

[coclass, progid(CTest.17), helpstring("interface CTest"), uuid(12341234-
1234-1234-1234-123412341234)]
struct CTest : ITest {
    void put_Grade(E e0, char c) {
        printf("CTest::put_Grade(e0 = %d,c = %c)\n", e0, c);
    }
    void get_Grade(E e0, char* pc) {
        printf("CTest::get_Grade(e0 = %d)\n", e0);
        *pc = 'A';
    }
    HRESULT Score(S* a, float b, VARIANT c) {
        printf("CTest::Score(a = %p,b = %f,c = %d)\n", a, b, c.iVal);
        return S_OK;
    }
};

```

FIGURE 6

```
#define _ATL_ATTRIBUTES 1
#include "atlbase.h"
#include "atlcom.h"
extern "C" int printf(const char*, ...);

#import "TestLib.tlb" embedded_idl ]

[project(name = MyLib, helpfile = "test.res", helpcontext = 12)];

[coclass, progid(CTest.17), helpstring("interface CTest"),
uuid(12341234-1234-1234-1234-123412341234)]
struct CTest : ITest {
    void put_Grade(E e0, char c) {
        printf("CTest::put_Grade(e0 = %d,c = %c)\n", e0, c);
    }
    void get_Grade(E e0, char* pc) {
        printf("CTest::get_Grade(e0 = %d)\n", e0);
        *pc = 'A';
    }
    HRESULT Score(S* a, float b, VARIANT c) {
        printf("CTest::Score(a = %p,b = %f,c = %d)\n", a, b, c.iVal);
        return S_OK;
    }
};
```

600

620

630

FIGURE 7

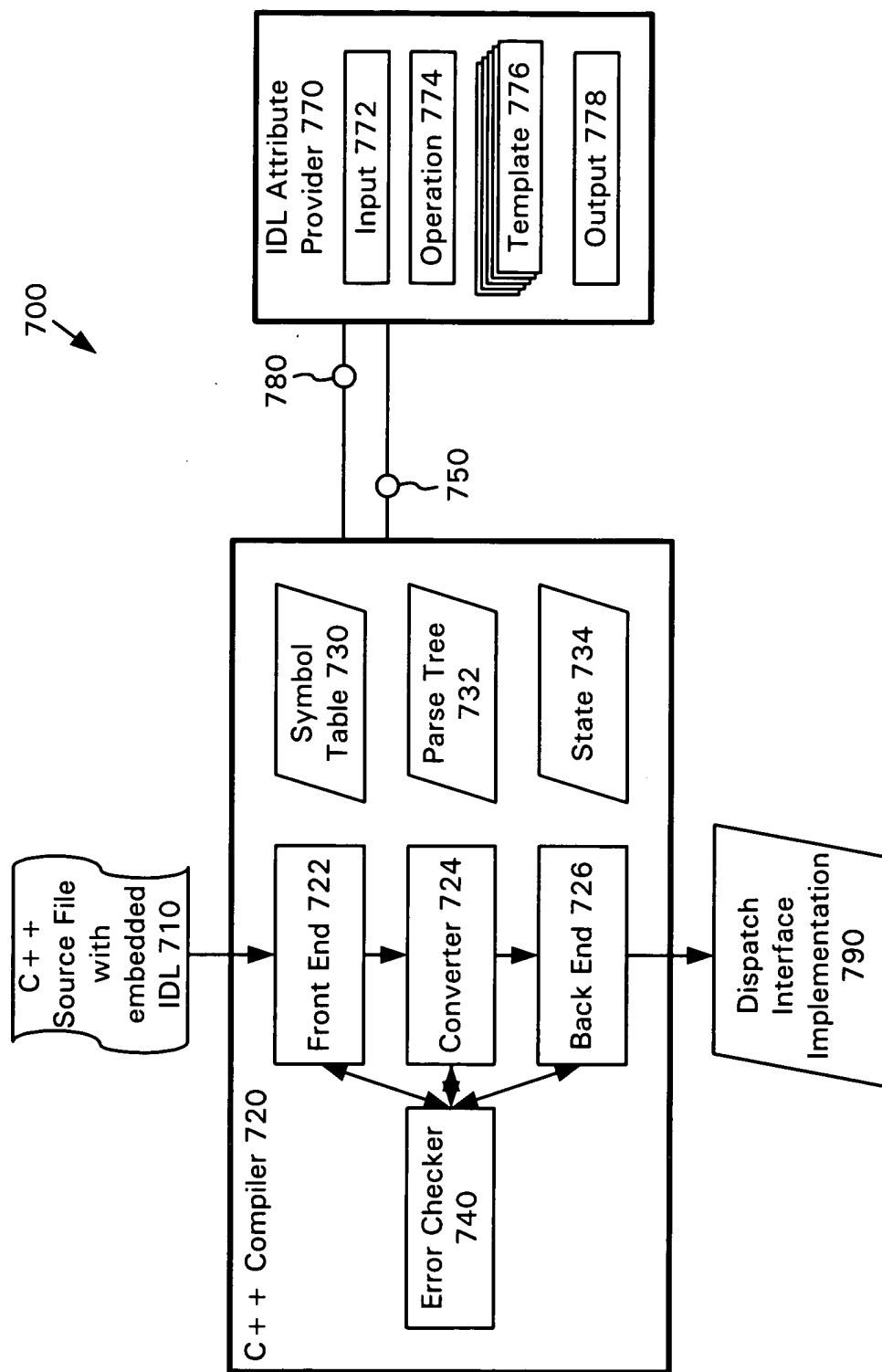


FIGURE 8

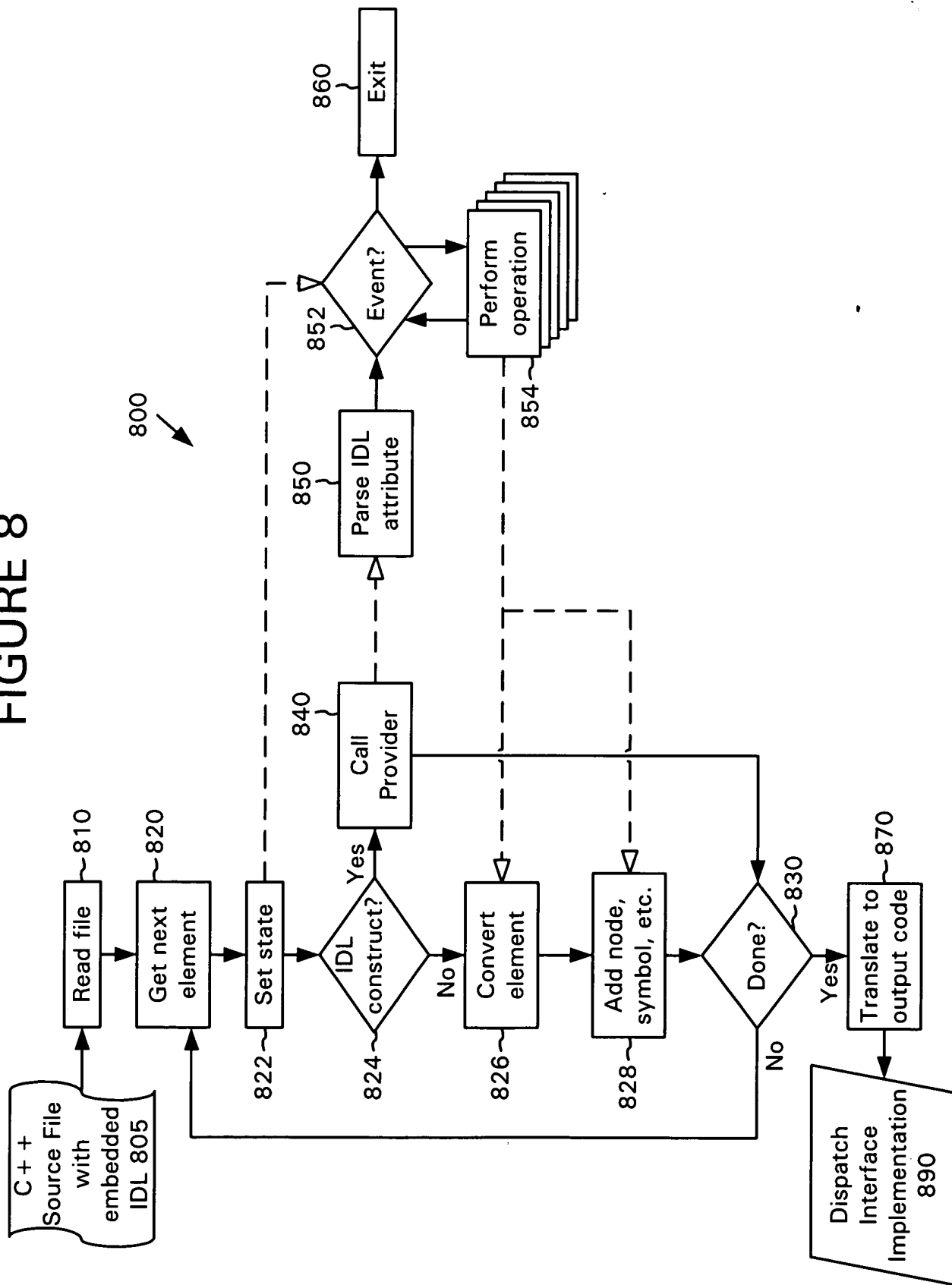


FIGURE 9a

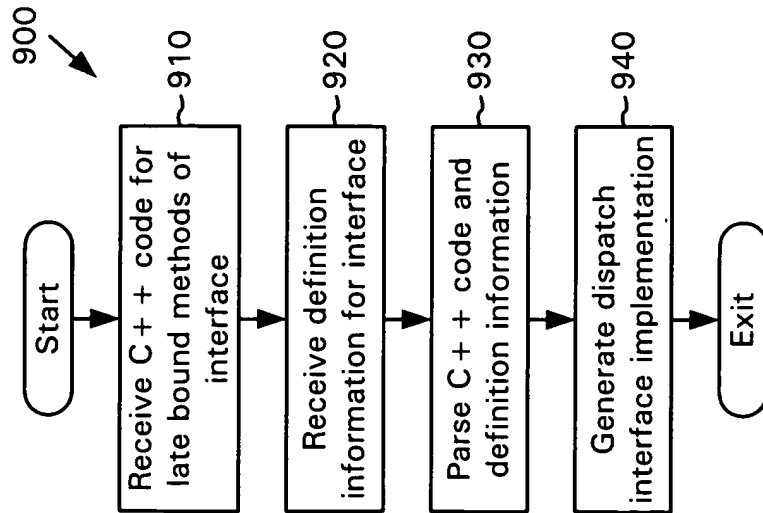


FIGURE 9b

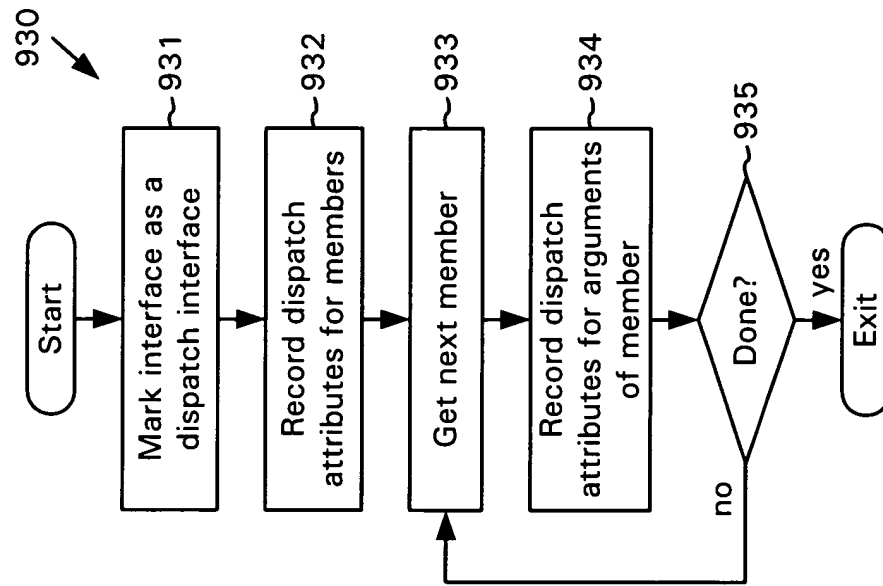


FIGURE 10a

```

1000
[
[coclass]
struct CTest : ITest ,
/* + + + Added Baseclass */ public CComCoClass<CTest, &__uuidof(CTest)> ,
/* + + + Added Baseclass */ public CComObjectRootEx<CComSingleThreadModel> ,
/* + + + Added Baseclass */ public IProvideClassInfoImpl<&__uuidof(CTest)>
{
    void put_Grade(E e0, char c) {
        printf("CTest::put_Grade(e0 = %d,c = %c)\n", e0, c);
    }
    void get_Grade(E e0, char* pc) {
        printf("CTest::get_Grade(e0 = %d)\n", e0);
        *pc = 'A';
    }
    HRESULT Score(S* a, float b, VARIANT c) {
        printf("CTest::Score(a = %p,b = %f,c = %d)\n", a, b, c.iVal);
        return S_OK;
    }
}
1010
]

```

To Figure 10b



```

1000 // + + + Start Injected Code
virtual HRESULT STDMETHODCALLTYPE Invoke(
    /* [in] */ DISPID dispidMember,
    /* [in] */ REFIID riid,
    /* [in] */ LCID lcid,
    /* [in] */ WORD wFlags,
    /* [out][in] */ DISPPARAMS *pDispParams,
    /* [out] */ VARIANT *pVarResult,
    /* [out] */ EXCEPINFO *pExcepInfo,
    /* [out] */ UINT *puArgErr)
{
    HRESULT hr = S_OK;
    if (pDispParams == 0) { return DISP_E_BADVARTYPE; }
    if (pVarResult != 0) { VariantInit(pVarResult); }
    switch (dispidMember) {
    case 18:
        {
            S* i1 = (S*) V_RECORD(&pDispParams->rgvarg[2]);
            float i2 = V_R4(&pDispParams->rgvarg[1]);
            VARIANT i3 = pDispParams->rgvarg[0];
            hr = Score(i1, i2, i3);
            if (pVarResult != 0) {
                V_VT(pVarResult) = VT_ERROR;
                V_ERROR(pVarResult) = hr;
            }
            break;
        }
    }
}
1020
1022
1024
1026
1028

```

To Figure 10a

To Figure 10c

FIGURE 10b

FIGURE 10c

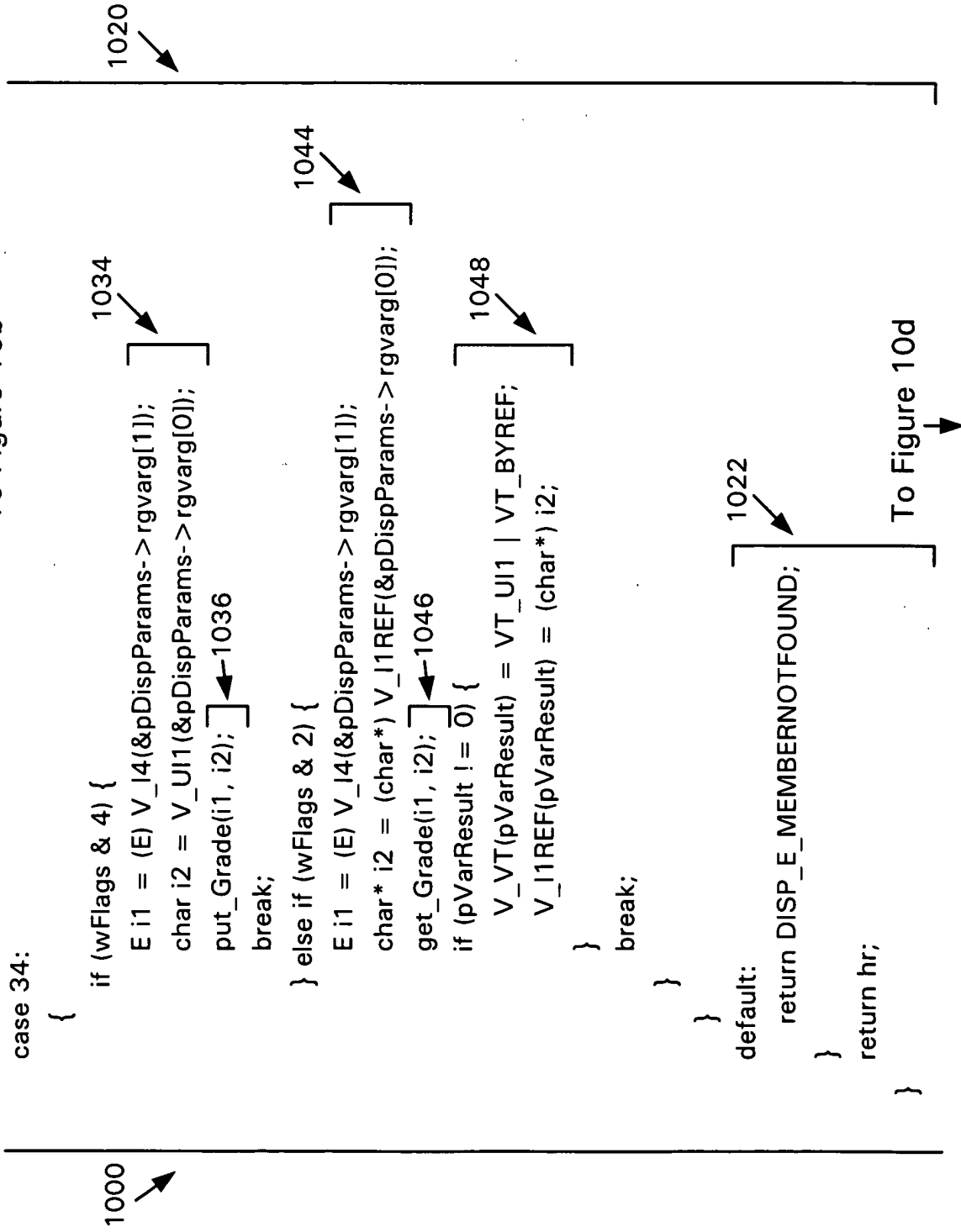


FIGURE 10d

↑ To Figure 10c

```
virtual HRESULT STDMETHODCALLTYPE GetIDsOfNames(  
    /* [in] */ REFIID riid,  
    /* [size_is][in] */ LPOLESTR *rgszNames,  
    /* [in] */ UINT cNames,  
    /* [in] */ LCID lcid,  
    /* [size_is][out] */ DISPID *rgDispld)  
{  
    static LPOLESTR names[] = { L"Grade", L"Score" };  
    static DISPID dids[] = { 34, 18 };  
    for (unsigned int i = 0; i < cNames; ++i) {  
        int fFoundIt = 0;  
        for (unsigned int j = 0; j < sizeof(names)/sizeof(LPOLESTR); ++j) {  
            if (lstrcmpW(rgszNames[i], names[j]) == 0) {  
                fFoundIt = 1;  
                rgDispld[i] = dids[j];  
            }  
        }  
        if (fFoundIt == 0) {  
            return DISP_E_UNKNOWNNAME;  
        }  
    }  
    return S_OK;  
}
```

↓ To Figure 10e

1000

1050

FIGURE 10e

↑
To Figure 10d

```

HRESULT TypeInfoHelper(REFIID iidDisp, LCID /*lcid*/, ITypeInfo** ppTypeInfo)
{
    if (ppTypeInfo == NULL) {
        return E_POINTER;
    }
    *ppTypeInfo = NULL;
    TCHAR szModule1[_MAX_PATH];
    ::GetModuleFileName(_pModule->GetModuleInstance(), szModule1, _MAX_PATH);
    USES_CONVERSION;
    CComPtr<ITypeLib> spTypeLib;
    HRESULT hr = LoadTypeLib(T2OLE(szModule1), &spTypeLib);
    if (SUCCEEDED(hr)) {
        CComPtr<ITypeInfo> spTypeInfo;
        hr = spTypeLib->GetTypeInfoOfGuid(iidDisp, &spTypeInfo);
        if (SUCCEEDED(hr)) {
            *ppTypeInfo = spTypeInfo.Detach();
        }
    }
    return hr;
}

```

1080
↓

↓
To Figure 10f

FIGURE 10f

↑
To Figure 10e

```

virtual HRESULT STDMETHODCALLTYPE GetTypeInfoCount(unsigned int* pctinfo)
{
    if (pctinfo == NULL) {
        return E_POINTER;
    }
    CComPtr<ITypelInfo> spTypelInfo;
    *pctinfo =
        (SUCCEEDED(TypelInfoHelper(__uuidof(ITest), 0, &spTypelInfo))) ? 1 : 0;
    return S_OK;
}

virtual HRESULT STDMETHODCALLTYPE GetTypeInfo(unsigned int iTInfo, LCID lcid, ITypelInfo** ppTInfo)
{
    if (iTInfo != 0) {
        return DISP_E_BADINDEX;
    }
    return TypelInfoHelper(__uuidof(ITest), lcid, ppTInfo);
}

//--- End Injected Code
};
    
```

1000 →

1060 →

1070 →

FIGURE 11a

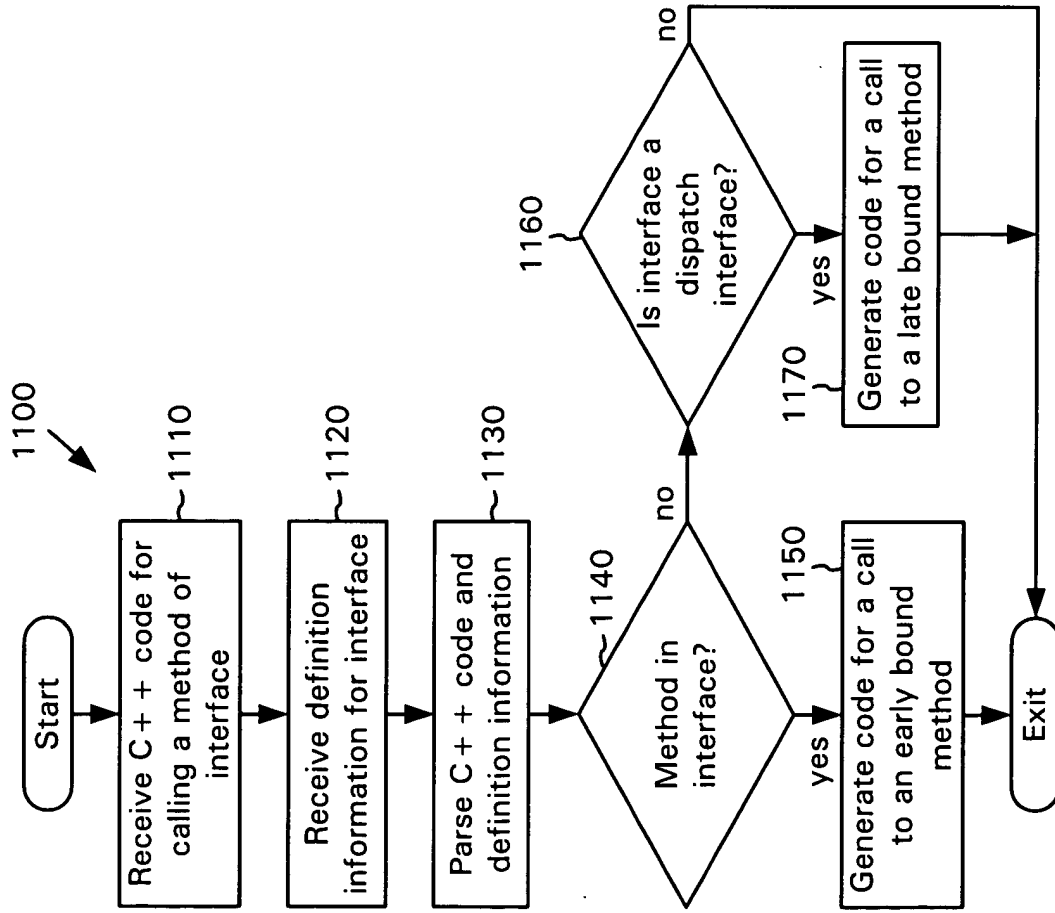


FIGURE 11b

